



Finalmente, el primer Servlet

Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas una vez más a este curso de servlets. Quería comenzar este video haciendo un poco de un repaso sobre cómo es que está funcionando todo este proyecto, cuando nosotros hacemos clic en ese botón verde para dar un run, ¿qué es lo que está ocurriendo por detrás?

[00:18] Lo que está ocurriendo es lo siguiente. Nosotros inicializamos la máquina virtual de Java que ella se encarga de ejecutar un método main que está dentro de una clase y, a través de eso, nosotros inicializamos nuestro Tomcat, nuestro servidor Tomcat que tiene su proyecto y tiene nuestro código HTM.

[00:40] ¿Entonces qué es lo que ocurre cuando nosotros hacemos una llamada a través de nuestro navegador para nuestro proyecto? Nosotros lo que hacemos es enviar una request http. Utilizamos el Protocolo HTTP, que el navegador lo conoce muy bien, y a través de él hacemos una request a nuestro servidor, que él contesta con un response para que el navegador lo lea.

[01:03] Entonces traduciendo eso a lo que nosotros hemos estado haciendo es utilizamos, enviamos como un parámetro todo esta URL dentro del request, que esa URL tiene el camino a dónde tenemos que apuntar, a dónde tenemos que enviarla bien, que es localhost, que es nuestra máquina, 8080 que es el puerto que puede ser modificado, es el por defecto.

[01:25] Gerenciador es el proyecto y `bienvenido.html` es el nombre de la página. Entonces enviamos todo eso y Tomcat reconoce cada uno de esos parámetros y dice: “Sí, gerenciador es un proyecto que yo tengo acá y `bienvenido.html` es una página”. Entonces lo que hace es devolver ese archivo HTML al browser, el navegador a través de una respuesta http, un response y el navegador interpreta ese HTML y se lo muestra al usuario.

[01:54] ¿Pero qué es lo que ocurre? Esto como está ahora en este momento es un contenido bastante estático, es un solamente un archivo HTML, entonces lo que nosotros queremos es darle algo de dinamismo. Lo que nosotros queremos es, por ejemplo que cuando recibimos una request poder enviar un email o poder también guardar datos en una base de datos o consultar una base de datos y a partir de los datos que obtengo, generar una página más dinámicamente.

[02:22] ¿Y cómo hago eso? Lo que voy a utilizar para hacer eso es un servlet. Este servlet es un objeto especial en el cual nosotros nos va a permitir, por ejemplo, modificar el nombre de la URL, que es como acá cambié, y a través de ese Protocolo HTTP yo no voy a necesitar llamar a una página HTML en particular, sino puedo simplemente colocar “hola” y eso va a producir toda esa maquinaria para poder devolver un archivo HTML dinámicamente.

[03:00] Bien, entonces, para eso sirve básicamente un servlet, para crear y devolver un contenido dinámico. Una curiosidad de dónde viene servlet. Bueno, ese let es en realidad un diminutivo en inglés, entonces sería algo así como un servidor pequeño o un servidorcito digamos.

[03:19] Tomcat es el servidor grande y servlet es un servidor pequeño. Entonces, una vez que tenemos esto, vamos a crear nuestro servlet. Para eso vamos a ir a nuestro Eclipse. Entonces, una vez que estamos en Eclipse, lo que nosotros queremos hacer es crear una clase, la cual va a ser la representación de este servlet, entonces nosotros vamos a hacer clic derecho, que es en ese `src/main/java`.

[03:47] Vamos a hacer new class. Vamos a dar un nombre al paquete para dejar todo mucho más normal, más padronizado. Entonces vamos a colocar un paquete com.alura.gerenciador.servlet. Solo para dejarlo bien organizado y en este caso vamos a crear algo llamado, por ejemplo, HolaMundoServlet.

[04:13] No necesita terminar con servlet, nada de eso. Es solamente a modo de ejemplo. Pero para que quede bien organizado vamos a usar ese nombre. Vamos a dar un finish y acá vemos que creó nuestra clase. Perfecto. Esta clase va a ser un Servlet. ¿Cómo va a ser un Servlet? Nosotros tenemos que extender extends de Servlet. ¿Pero cuál Servlet?

[04:43] Bueno, en este caso es un HttpServlet, va a atender a las consultas, los envíos de Http, entonces, a través del Protocolo Http, eso quise decir. Entonces, vamos a colocar HttpServlet. Y vean que está en rojo, porque nos está diciendo que nosotros tenemos que importar ese HttpServlet. Entonces vamos a hacer eso, simplemente colocando el mouse acá arriba, hacemos clic en import. Perfecto.

[05:11] Ya nos agregó el import y acá nos puso un mensaje en amarillo, pero no se preocupen, eso pueden ignorarlo, ignorarlo tranquilamente o bien este servlet, ¿qué es lo que va a hacer? Este Servlet va a tener que servir, va a tener que recibir las requisiciones para poder enviar una respuesta. Entonces lo que vamos a tener que hacer es llamar un método llamado service.

[05:40] ¿Ese service qué es lo que nosotros vamos a tener que hacer? Ese service pertenece a ese HttpServlet. Entonces nosotros lo que podemos hacer es control y barra espaciadora, y acá nos trae dos opciones. La que a nosotros nos interesa es la primera. ¿Por qué? Porque esta tiene una request y va a tener la response.

[06:01] O sea lo que nosotros vamos a recibir, los datos que vamos a recibir y los datos que tenemos que enviar, entonces vamos a hacer doble clic acá, no se asusten, tienen muchas cosas acá, pero vamos a simplificarlo. Primero voy a

colocar ese `HttpServletRequest`, vamos a cambiarle el nombre para que quede más bonito.

[06:23] Vamos a colocarle `req`, que es para simplificar `request` y está `response` vamos a colocar `resp`, solo para tener las referencias bastante más simples. En este caso, estas excepciones nosotros las podemos eliminar ahora, solo para dejarlo más simple, eliminar esas `exceptions` es mucho más fácil que agregarlas.

[06:47] Esta parte de `super` nosotros no vamos a llamar al método padre de `service`, simplemente vamos a eliminarlo. Y este mensaje con “Ctrl + D” también vamos a eliminarlo. Perfecto, tenemos nuestro método, ese `Override` puede quedar ahí, porque nosotros vamos a modificar el método de nuestra clase padre para atender a lo que nosotros queremos hacer. Bien. Ya tenemos ese método.

[07:17] Ahora, ¿qué es lo que tenemos que hacer? Nosotros queremos hacer dos cosas, queremos cambiar en la URL esa parte final para que `hola` sea nuestro Servlet, reconocer que ese Servlet es el `/hola` dentro de la URL, y segundo enviar una respuesta en la `response` a través del Protocolo `Http`.

[07:39] Entonces para eso vamos a utilizar una cosa. Acá voy a colocar esto, solo como para dejar una anotación para recordar que tenemos que hacer eso. Y vamos a utilizar una cosa que se llaman anotaciones. Recuerdan que les dije que teníamos que utilizar un Tomcat con una versión 7 o superior. Bueno, es justamente por estas anotaciones.

[08:01] Estas anotaciones lo que hacen es crear código en el `building` que nos ayuda a hacer algunas conexiones entre la máquina virtual y nuestro código que estamos colocando, entonces en este caso esas anotaciones se utilizan para crear una anotación.

[08:23] Utilizamos el `@` y en este caso nosotros la anotación que queremos es la `@WebServlet`. Con “Ctrl + espacio” ya nos hace el `import` automáticamente. Y

dentro de esa `webServlet`, de esa anotación `webServlet`, va a recibir un parámetro. ¿Cuál parámetro va a ser ese?

[08:45] Bueno, con “Ctrl + espacio” vemos todas las cosas que podemos adicionar como parámetro, y en este caso nosotros queremos un URL pattern. URL pattern va a ser lo que nosotros vamos a agregar acá en el final del nombre que va a llamar a nuestro Servlet en la URL.

[09:05] Entonces lo que vamos a hacer es con “Ctrl + espacio”, colocamos `urlPatterns`, doble clic, nos auto completa y con comillas nosotros vamos a colocar ese “/hola” que habíamos puesto en la presentación. Entonces acá pueden poner lo que ustedes quieran, no necesita llamarse `HolaMundo`, nada por el estilo, solamente voy a poner “/hola” para tener un nombre sencillo y que podamos llamarlo fácilmente.

[09:35] Bien. Ya hicimos la primera parte de las dos que teníamos que hacer. Ahora vamos a ir con la segunda. La segunda es enviar una página web para el navegador, ¿cómo vamos a hacerlo? Vamos a necesitar utilizar uno de estos parámetros. Tenemos el `request` y el `response`, entonces lo que vamos a utilizar es el `response`, porque es la respuesta a la que nosotros queremos enviar.

[09:58] ¿Qué vamos a hacer con ese objeto `response`? En el mundo `http`, nosotros podemos devolver dos tipos de datos básicamente. Tenemos el `string` de datos, por ejemplo, se utiliza cuando tenemos que enviar un PDF que es todo ya un archivo o una imagen, por ejemplo, una imagen es un `string` de datos.

[10:20] Tenemos esa opción y tenemos también, por ejemplo, enviar HTML directamente. Entonces, por ejemplo, si nosotros quisiéramos enviar un `string` de datos, nosotros utilizaremos este método de `getOutputStream`. Pero ese no es el caso, no es lo que nosotros queremos hacer en este momento. Nosotros queremos enviar un contenido HTML.

[10:42] Entonces vamos a usar un `getWriter` acá para facilitar mi vida, vamos a colocar “Ctrl + espacio”, `getWriter` y vean lo siguiente. Este método `getWriter` voy a colocar de nuevo acá `getWriter`. Veán que devuelve un objeto del tipo `PrintWriter`.

[11:10] Eso nos va a ayudar para identificar lo que tenemos que hacer después, cuando tenemos ese `getWriter`, nosotros lo que podemos hacer es decir que vamos a tener un objeto del tipo `PrintWriter`, podemos llamarlo `out` y va a ser igual a esto, este método. Como nos está diciendo que está en rojo porque no hemos importado ese ese tipo de dato, entonces vamos a usar ese `import`.

[11:37] ¿Y qué es lo que nos está diciendo acá en rojo? Nos está diciendo que necesitamos agregar una `throw exception` dentro de nuestro método. Entonces simplemente vamos a hacer clic acá en `throw exception`. Como ya nos está pidiendo el método que lo hagamos, lo vamos a hacer.

[11:53] Bien, nosotros tenemos nuestro objeto por el cual nosotros vamos a imprimir los datos que queremos que el navegador reciba. ¿Ahora, cómo vamos a utilizarlo? Simplemente vamos a utilizar ese objeto llamado `out`. y lo que nosotros vamos a hacer es un `println`.

[12:12] No se confundan con el `system.out.println` que imprime en una consola que se ve en Java. Este `println` lo que vas a hacer es imprimir en ese stream el cual nosotros vamos a enviar para el navegador. Entonces, dentro de ese `println` nosotros vamos a enviar nuestra respuesta HTML.

[12:32] Vamos acá a abrir unas llaves para comenzar a escribir “`html`” y para facilitar mi vida usando “Ctrl + Alt + flecha para abajo” y vamos a repetir esas líneas para hacer todo mucho más rápido. Tenemos ese “`html`”, nosotros tenemos un `body` también, acá va a ir nuestro mensaje y acá vamos a cerrar las etiquetas “`/body`” y “`/html`”.

[12:59] En este mensaje nosotros vamos a poner: “Hola Mundo! Felicitaciones por crear tu primer Servlet!” Perfecto. Ya con esto, lo que nosotros estamos

haciendo entonces es imprimir en la respuesta, en la response Http estamos colocando directamente una página.

[13:25] Esto aún no es muy dinámico, pero ya es un avance en relación a una página html pura. Entonces, una vez que hemos hecho esto vamos a hacer un “Ctrl + S” para guardar y acá vamos a hacer clic en el servidor y vamos a correr, colocar un run. Vamos a ver en la consola si está todo bien, Server startup, bien, no hay ningún mensaje de error.

[13:53] Siempre es bueno ver la consola en caso de que haya algún error. En teoría, si nosotros vamos a la página /hola, gerenciador/hola, vean que llegó nuestro mensaje. En el caso de que a ustedes no les haya funcionado, no haya llegado, fíjense primero si está todo correcto dentro de lo que es el HTML, también para el HTML pueden hacer clic derecho, ver código fuente de página y ver si acá llegó bien el HTML o no.

[14:27] Veán si está todo bien, las etiquetas, los nombres, y por ejemplo, una cosa que podemos hacer nosotros como desarrolladores es acá hacer un `system.out.println`, este sí va a imprimir en la consola, y vamos a colocar acá entre comillas “Se ejecutó el servlet”.

[14:57] Eso es solo para como un mensaje para nosotros programadores, “Ctrl + S”, vamos a ejecutar de nuevo el servidor por las dudas. A veces no es necesario, a veces automáticamente hacer un restart, pero por las dudas vamos a ejecutarlo. Vamos a dar un enter, vean que acá, enter.

[15:20] No ha cambiado el mensaje, pero si nosotros vamos acá, vean que acá aparece dos veces porque ejecuté dos veces la llamada, pero entonces con este mensaje nosotros vemos que realmente entró dentro de ese, cuando nosotros hicimos la llamada, entró dentro de ese método y se ejecutó y realmente imprimió ese mensaje.

[15:41] Entonces significa que hasta ahí está todo bien. Perfecto. Ya hemos creado nuestro primer Servlet, entonces ahora hagan los ejercicios y los espero

en el próximo video.